

AMENDMENTS TO THE CLAIMS

1. (Currently amended) A computer-implemented method of ~~sharing markup text from a page among a plurality of users in response to a requests from said plurality of users~~dynamically generating web pages, said method comprising:
 - analyzing the ~~a page to extract markup text, wherein the page that~~ a page to extract markup text, wherein the page ~~includes at least the markup text and a set of code instructions executable to be executed on a server as an application;~~ includes at least the markup text and a set of code instructions executable to be executed on a server as an application;
 - ~~pre-initializing a static variable of a class to contain~~ extracting the markup text from the page; and
 - generating a servlet class for the page based on the set of code instructions, wherein the servlet class does not include the markup text;
 - ~~loading the class containing the pre-initialized static variable into a shared, read-only memory, wherein the shared, read-only memory is accessible to said plurality of users;~~ a copy of the markup text into shared memory;
 - in response to each request of a plurality of requests for the page from a plurality of clients, performing the steps of
 - instantiating a distinct instance of the servlet class on the server, wherein instantiating each instance of the servlet class does not create another copy of the markup text;
 - executing said distinct instance of the servlet class, wherein execution of each instance of the server class generates a compiled page based on the copy of the markup text that resides in shared memory, and the set of code instructions; and

sending the compiled page to a client that requested the page.

2. (Previously Presented) The computer-implemented method according to claim 1, further comprising storing the markup text in a resource file associated with the application.
3. (Canceled)
4. (Canceled)
5. (Currently amended) A method of initiating a first instance of an application that shares a set of markup text with other instances of the application, wherein the first instance of the application is generated by compiling code from a page that contains both the code and the set of markup in response to a request from one or more users, said method comprising:
executing instructions to instantiate the first instance of the application, wherein
said instructions are stored on a computer-readable medium, said
instructions that, when executed, cause one or more processors to
perform the steps of:
analyzing the page to ~~distinguish~~ extract ~~the code and~~ the set of markup
text;
~~pre-initializing a static variable of a class to contain the markup text from~~
the page; and

generating a servlet class for the page based on the code from the page,
wherein the servlet class does not include the markup text;
~~loading the class containing the pre-initialized static variable into a~~
~~shared, read-only memory, wherein the shared, read-only~~
~~memory is accessible to the one or more users; a copy of the~~
markup text into shared, read-only memory;
in response to each request of a plurality of requests for the page from
the one or more users, performing the steps of
instantiating a distinct instance of the servlet class, wherein
instantiating each instance of the servlet class does not
create another copy of the markup text;
executing said distinct instance of the servlet class, wherein
execution of each instance of the server class generates a
compiled page based on the copy of the markup text that
resides in shared, read-only memory, and the set of code
instructions;
sending the compiled page to a client that requested the page; and
accessing the set of markup text in the shared, read-only memory when
the code from the first instance of the application is executed.

6. (Previously Presented) A method according to claim 5, wherein the class is not loaded into the shared, read-only memory when the other instances of application are executed.

7. (Canceled)
8. (Canceled)
9. (Currently amended) A computer-implemented method according to claim 1,
wherein:
the markup text includes information to be displayed to a user and an annotation
directing a user agent how to render the information to be displayed to the
user; and
the markup output by the executing servlet class ~~application~~ includes the
annotation.
10. (Canceled)
11. (Previously Presented) A method according to claim 5, wherein:
the set of markup text includes information to be displayed to a user and an
annotation directing a user agent how to render the information to be
displayed to the user; and
the set of markup output by the application includes the annotation.
12. (Canceled)
13. (Canceled)

14. (Canceled)
15. (Currently amended) A computer-readable storage medium bearing instructions that, when executed, cause one or more processors to perform a method for sharing markup text from a page among a plurality of users in response to requests from said plurality of users, said method comprising:
- ~~analyzing the a page to extract markup text, wherein the page that includes at least the markup text and a set of code instructions executable to be executed on a server as an application;~~
- ~~pre initializing a static variable of a class to contain extracting the markup text from the page; and~~
- ~~generating a servlet class for the page based on the set of code instructions, wherein the servlet class does not include the markup text;~~
- ~~loading the class containing the pre initialized static variable into a shared, read-only memory, wherein the shared, read-only memory is accessible to said plurality of users. a copy of the markup text into shared memory;~~
- ~~in response to each request of a plurality of requests for the page from a plurality of clients, performing the steps of~~
- ~~instantiating a distinct instance of the servlet class on the server, wherein instantiating each instance of the servlet class does not create another copy of the markup text;~~
- ~~executing said distinct instance of the servlet class, wherein execution of each instance of the server class generates a compiled page based~~

on the copy of the markup text that resides in shared memory, and
the set of code instructions; and
sending the compiled page to a client that requested the page.

16. (Previously Presented) The computer-readable storage medium of claim 15, further comprising instructions to store the markup text in a resource file associated with the application.
17. (Currently amended) The computer-readable storage medium of claim 15, wherein:

the markup text includes information to be displayed to a user and an annotation directing a user agent how to render the information to be displayed to the user; and

the markup output by the executing servlet class application includes the annotation.
18. (Canceled)
19. (Currently amended) A computer-readable storage medium bearing instructions that, when executed, cause one or more processors to perform a method of initiating a first instance of an application that shares a set of markup text with other instances of the application, wherein the first instance of the application is generated by compiling code from a page that contains both the code and the set

of markup text in response to a request from one or more users, said method comprising:

executing instructions to instantiate the first instance of the application;

wherein the instructions to instantiate the first instance of the application

include:

analyzing the page to distinguish extract the code and the set of markup text;

~~pre-initializing a static variable of a class to contain the markup text from the page; and~~

generating a servlet class for the page based on the code from the page,

wherein the servlet class does not include the markup text;

~~loading the class containing the pre-initialized static variable into a shared, read-only memory, wherein the shared, read-only memory is accessible to the one or more users; a copy of the markup text into shared, read-only memory;~~

in response to each request of a plurality of requests for the page from

the one or more users, performing the steps of

instantiating a distinct instance of the servlet class, wherein

instantiating each instance of the servlet class does not

create another copy of the markup text;

executing said distinct instance of the servlet class, wherein

execution of each instance of the server class generates a

compiled page based on the copy of the markup text that

resides in shared, read-only memory, and the set of code instructions;

sending the compiled page to the user that requested the page; and
accessing the set of markup text in the shared, read-only memory when the code
from the first instance of the application is executed.

20. (Previously Presented) The computer-readable storage medium of claim 19,
wherein the class is not loaded into the shared, read-only memory when the other
instances of the application are executed.

21. (Previously Presented) The computer-readable storage medium of claim 19,
wherein:
the set of markup text includes information to be displayed to a user and an
annotation directing a user agent how to render the information to be
displayed to the user; and
the set of markup output by the application includes the annotation.

22. (Canceled)

23. (New) A computer-implemented method according to claim 1, wherein the servlet
class includes an inner class.

24. (New) A computer-implemented method according to claim 23, wherein the step of loading a copy of the markup text includes hot-loading an instance of the inner class.
25. (New) A computer-implemented method according to claim 24, wherein the inner class comprises an array of characters.
26. (New) A method according to claim 5, wherein the servlet class includes an inner class.
27. (New) A method according to claim 26 wherein the step of loading a copy of the markup text includes hot-loading an instance of the inner class.
28. (New) A method according to claim 27, wherein the inner class comprises an array of characters.
29. (New) A computer-readable storage medium according to claim 15, wherein the servlet class includes an inner class.
30. (New) A computer-readable storage medium according to claim 29, wherein the step of loading a copy of the markup text includes hot-loading an instance of the inner class.

31. (New) A computer-readable storage medium according to claim 30, wherein the inner class comprises an array of characters.
32. (New) A computer-readable storage medium according to claim 19, wherein the servlet class includes an inner class.
33. (New) A computer-readable storage medium according to claim 32, wherein the step of loading a copy of the markup text includes hot-loading an instance of the inner class.
34. (New) A computer-readable storage medium according to claim 33, wherein the inner class comprises an array of characters.